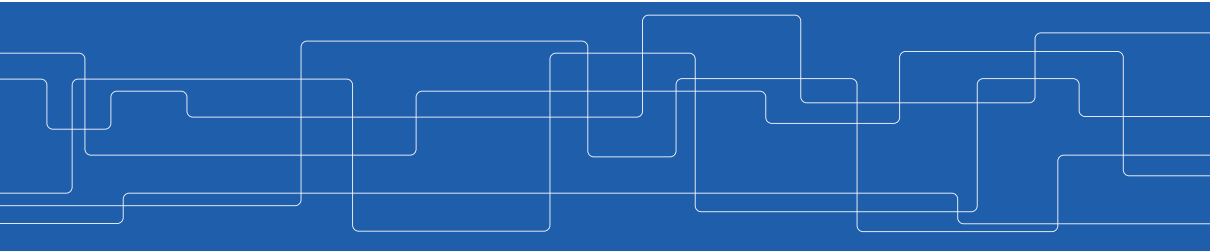
Introduction

Jim Dowling
jdowling@kth.se
Course Assistants: Fabien Schmidt, Tianze Wang
2022-10-31
Slides by Amir H. Payberah



Course Information

Course Objective

- This course has a system-based focus.
- Learn the theory of machine learning and deep learning.
- Learn the practical aspects of building machine learning and deep learning algorithms using data parallel programming platforms, such as Spark and TensorFlow.


KTH
VETENSKAP
OCH KONST

Topics Covered in the Course

- Part 1: large scale deep learning
  - TensorFlow
  - Deep Neural Networks (DNN)
  - Different DNN architectures, e.g., CNNs, RNNs, Autoencoders, GAN
  - Distributed learning

- Part 2: large scale machine learning
  - Serverless Machine Learning
  - MLOps (machine learning operations)
  - Feature Stores for Machine Learning
  - Feature Engineering at Scale

Intended Learning Outcomes (ILOs)

- ILO1: explain the principles of ML/DL algorithms and apply their techniques to solve problems.
- ILO2: demonstrate an ability to design DNN architectures and explain challenges to scaling their training and inference with increasing compute and data.
- ILO3: explain the principles of scaling machine learning systems.
- ILO4: implement scalable ML/DL systems.

The Course Assessment

- Task1: the Examination on 16th January 2023(A-F)
- Task2: the lab assignments (A-F)
- Task3: the final project (P/F)

How Each ILO is Assessed?

	Task1	Task2	Task3
ILO1	x		
ILO2	x	x	x
ILO3	x	x	
ILO4		x	x

Task1: The Exam (A-F)

- Questions about the lectures and the labs.
- Some sample questions for the exam will be distributed in mid December.
- The examination is graded (A-F).

Task2: The Lab Assignments (A-F)

- Two lab assignments: source code and oral presentation.
- E: source code
- D: source code + half questions (basic)
- C: source code + all questions (basic)
- B: source code + half questions (basic and advanced)
- A: source code + all questions (basic and advanced)

Task3: The Final Project (A-F)

- One final project: source code and oral presentation.
- Proposed by students and confirmed by the teacher: A-level or C-level proposals.
- E: C-level source code
- D: C-level source code + half questions (basic and advanced)
- C: C-level source code + all questions (basic and advanced) or A-level source code + all questions (basic)
- B: A-level source code + half questions (basic and advanced)
- A: A-level source code + all questions (basic and advanced)

The Final Grade

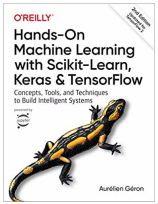
- The final grade is the weighted average of the Exam (0.3), two labs (0.15 each), and the final project (0.4).
- To compute it, map A-E to 5-1, and take the average.
- The floating values are rounded up, if they are more than half, otherwise they are rounded down.
  - E.g., 3.6 will be rounded to 4, and 4.5 will be rounded to 4.
- A late submission will reduce you grade level by one. That is, A will become B, B will become C, and so on.
- To pass the course, you need to take at least E in all the assignments.

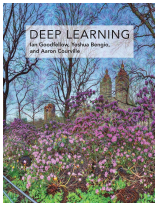
How do you submit the Assignments?

- Through the Canvas site.
- Students will work in groups of two on Task 2 and Task 3.

The Course Material

- Hands-on machine learning with Scikit-Learn and TensorFlow, 2nd Edition, A. Geron, O'Reilly Media, 2019
- Deep learning, I. Goodfellow et al., Cambridge: MIT press, 2016





The Course Web Page

https://id2223kth.github.io

Supervised Machine Learning 101

Features in Machine Learning

Features are properties of things you want to use to make predictions

*For example, predict if a fruit **is either an apple or an orange** based on its color/weight.*

feature color: green
feature weight: 70-250gr

feature color: orange
feature weight: 60-300gr

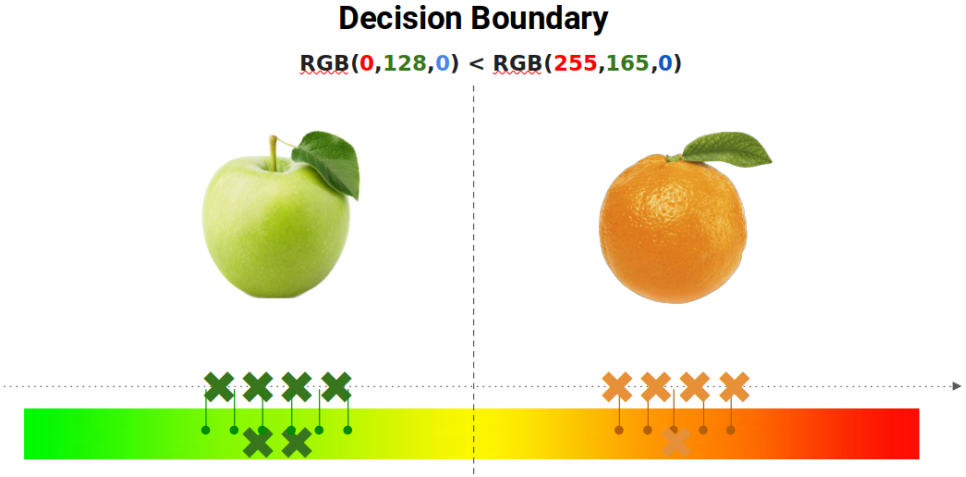
Supervised Machine Learning (ML)

Each example has both the features (the fruit's color) and a label that is either "apple" or "orange". The **label** is the 'target' we are trying to predict, using the input **features** (the fruit's color).

Examples of Apples

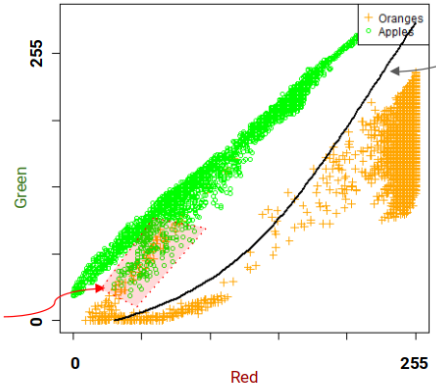
Example of Oranges

A Linear Model can classify the Fruit (Classifier)



The Decision Boundary

For a large enough data set, even with a decision tree classifier and 2 features, **it's still not 100% accurate**



Decision Boundary

Hard to correctly classify these examples ("outliers") as apples or oranges.

Note: we have 2 features - 'red' and 'green' colors, range 0-255. **We ignore the "blue" channel as it is 0 for both apples and oranges.**

Source code for a small sample of data with a decision tree classifier

```
import sklearn
from sklearn import tree
# 4 examples of features with [red-color, green-color]
features = [[0,120], [0, 110], [250, 150], [255, 163]]
# green apples == 0; oranges == 1
labels = [0, 0, 1, 1]

clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)

test_fruits = [[0,128], [249, 155]]
test_labels = [0, 1]
pred_labels = clf.predict(test_fruits)
print(pred_labels)
```

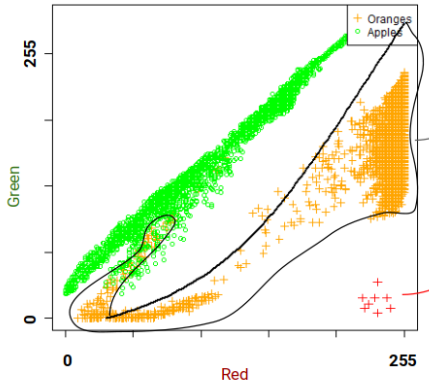
But wait, apples can also be red!

There is no straight-line decision boundary based on 2 color channels (red and green)

RGB(0,128,0) RGB(255,165,0) RGB(255,0,0)

It's harder to separate Green Apples, Oranges, and Red Apples with just 2 colors (red and green)

We need a non-linear classifier to learn to separate apples and oranges based on our 2 color channels (we don't need the blue channel)



Good decision boundary, but it's harder to learn

Red apples

Note: we have 2 features - 'red' and 'green' colors; 'blue' color is the same (0) for both apples and oranges.

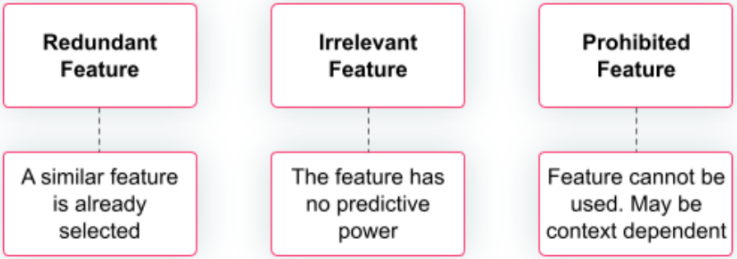
From two dimensions to millions of dimensions

- Can we just add more features? Yes. If you add weight and smoothness, you can separate apples and oranges. We could plot our fruit in 3d or even 4d and find a plane that separates apples and oranges

- You can add many more features (dimensions). With the caveat that too many dimensions can lead to overfitting. Overfitting means your model is not good at generalizing to correctly predict new fruit examples (it would work well for the training data, but not unseen (new) examples)

- In image classification, each pixel is a feature. That's millions of features for a single HD image. Deep learning can be used to train models with millions of features.

Not all properties with predictive power should be features



With current inadequate AI legislation/norms, you are personally responsible to build ethical AI systems

Lots of features: deep learning for apple classification works

Ayaz et al show how deep learning and image classifiers **can identify rotten/blotched/scab apples,** shown here.

Deep learning models **as non-linear classifiers**

To work well, they need **lots of labelled training data and GPUs** to train models on

Figure 6. Scab apple process through binary thresholding, clustering and DCNN.

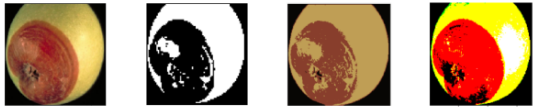


Figure 7. Rot apple process through binary thresholding, clustering and DCNN.

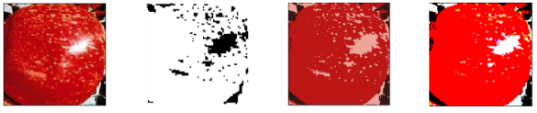


Figure 8. Blotch apple process through binary thresholding, clustering and DCNN.

A model can also be trained to predict a number (regression)

A regression problem:
train an ML model to estimate the weight of an apple given its dimensions (diameter) and color.

(red, 9cm) -> 120gr

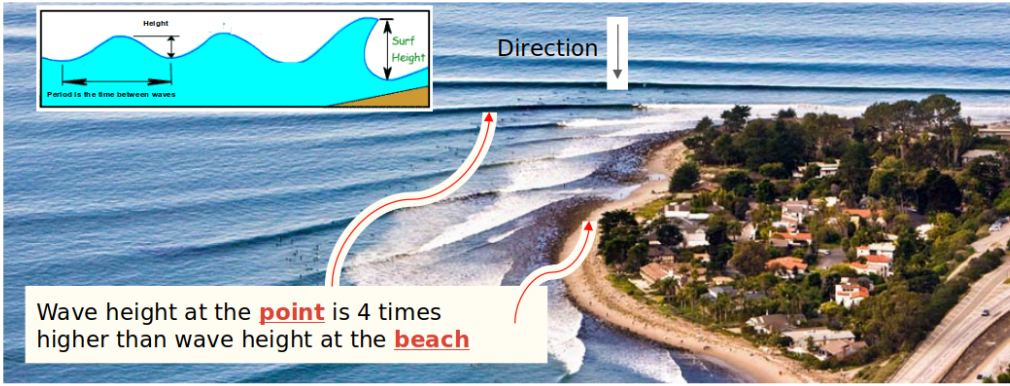
(dark green, 11cm) -> 180gr

(light green, 13cm) -> 160gr

Predicting Surf Height at a beach – Classification or Regression?

Classification or regression?

Problem: predict the height of surf at a beach (see example system on the course website), three features that are useful are: **(swell height, period, direction)** -> **Surf Height in feet**



What is Supervised Machine Learning, then?

- With our Apple/Orange classifier, we used (features, label) examples to train a model to find a decision boundary.
- Then when a new fruit arrived, we could use the model to predict if the new piece of fruit is an apple or an orange.
- We can generalize to say that supervised machine learning is concerned with:
- extracting a pattern from labeled data (features) to a model
- using that model to make predictions for new unlabeled data (features)

Tradtional ML courses vs ID2223

- Static Datasets, where Features for ML are correct and unbiased
- The goal is to optimize your model with a model evaluation metric (accuracy) to communicate the value of your model

---

- Data never stops coming and it comes from heterogeneous data sources
- Communicate the value of your model as a Prediction Service - one that can be scaled and deployed using MLOps (Machine Learning Operations) best practices (versioning, automated testing/deployment)

Feature Engineering is often treated like this “helpful” guide to drawing a Barn Owl

How to Draw an Owl:

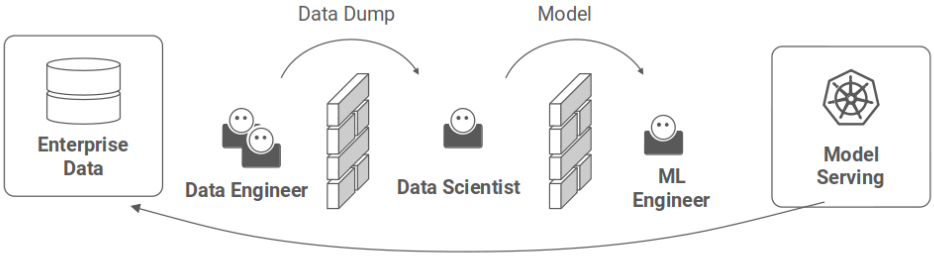
1. Draw Circles

2. Now, “simply” draw the rest of the Owl

Eextract the features from the input data. We will study feature engineering at scale.

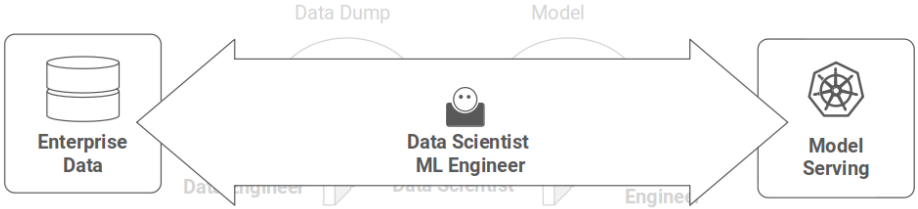
Raw Data	**Extracted Feature**	**Method**
Hotel room bookings	Weekly vacancy level	Aggregation
User's web session history	Session history embedding	Dimensionality Reduction
User's date of birth		Binning
Hourly spot electricity prices	Scale into range [0, 1]	Normalization
User's home country	Binary number of country	One Hot Encoding

Many Enterprises have walled gardens between teams building production ML Systems



In many large organizations,
data scientists only build and evaluate models

With infrastructure support (Serverless Machine Learning), developers need less infrastructural skills to deploy ML Systems



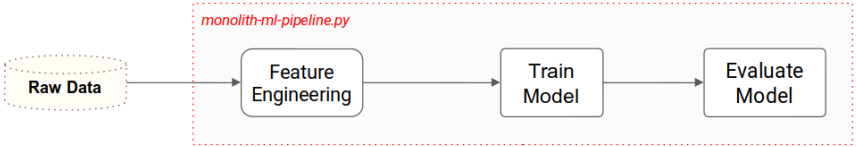
Don't stop at training a model, build its prediction service

Learn to deploy your models, evaluate, and debug them.

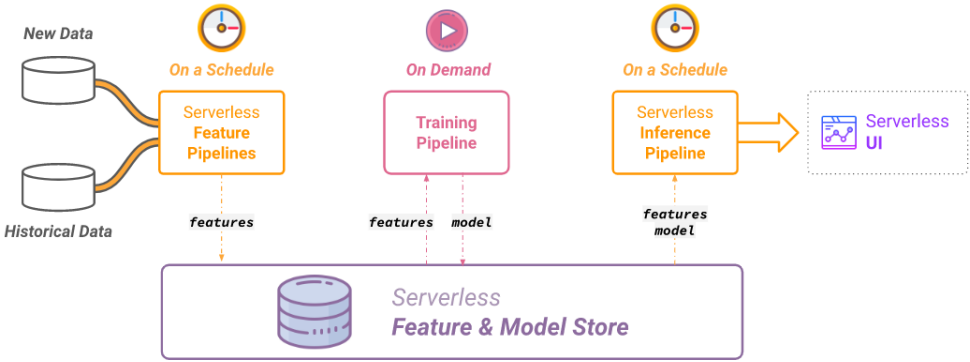
Leverage **Serverless Machine Learning** to avoid installing and managing the infrastructure needed to machine learning in production.

Monolithic ML Pipeline

- A pipeline is a program that takes and input and produces an output
- End-to-end ML Pipelines are a single pipeline that transforms raw data into features and trains and scores the model in one single program



Refactor the Monolithic ML Pipeline to Scale your ML Systems



Machine Learning and Deep Learning

Learning Algorithms

- A ML algorithm is an algorithm that is able to learn from data.
- What is learning?
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (Tom M. Mitchell)

Learning Algorithms - Example 1

- A spam filter that can learn to flag spam given examples of spam emails and examples of regular emails.
- Task T: flag spam for new emails
- Experience E: the training data
- Performance measure P: the ratio of correctly classified emails

[https://bit.ly/2oiplYM]

Learning Algorithms - Example 2

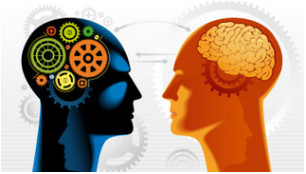
- Given dataset of prices of 500 houses, how can we learn to predict the prices of other houses, as a function of the size of their living areas?
- Task T: predict the price
- Experience E: the dataset of living areas and prices
- Performance measure P: the difference between the predicted price and the real price

[https://bit.ly/2MyiJUy]

Types of Machine Learning Algorithms

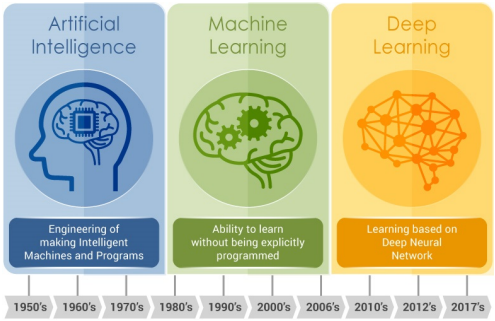
- Supervised learning
  - Input data is labeled, e.g., spam/not-spam or a stock price at a time.
  - Regression vs. classification

- Unsupervised learning
  - Input data is unlabeled.
  - Find hidden structures in data.



AI Generations - Deep Learning

- For many tasks, it is difficult to know what features should be extracted
- Use machine learning to discover the mapping from representation to output



[https://bit.ly/2woLEzs]

Image Classification with Deep Learning

- For image classification, where each pixel is a feature, Deep Learning can do the feature extraction as part of the learning algorithm.

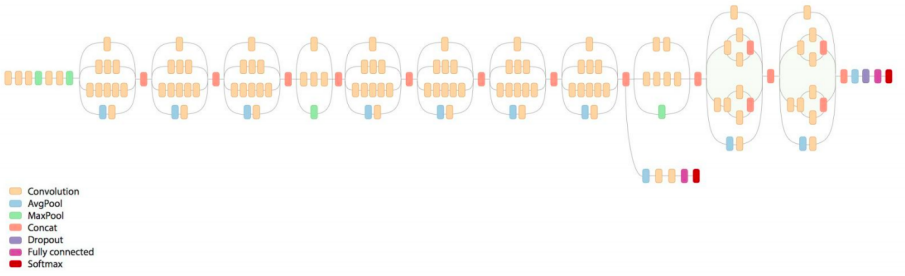
Sheepdog or Mop

Chihuahua or Muffin

Barn Owl or Apple

Training Deep Neural Networks

- Computationally intensive
- Time consuming

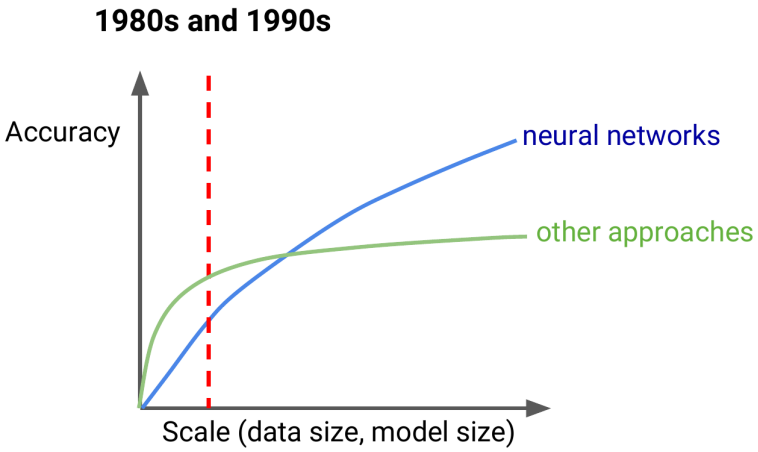


[https://cloud.google.com/tpu/docs/images/inceptionv3onc--oview.png]

Why?

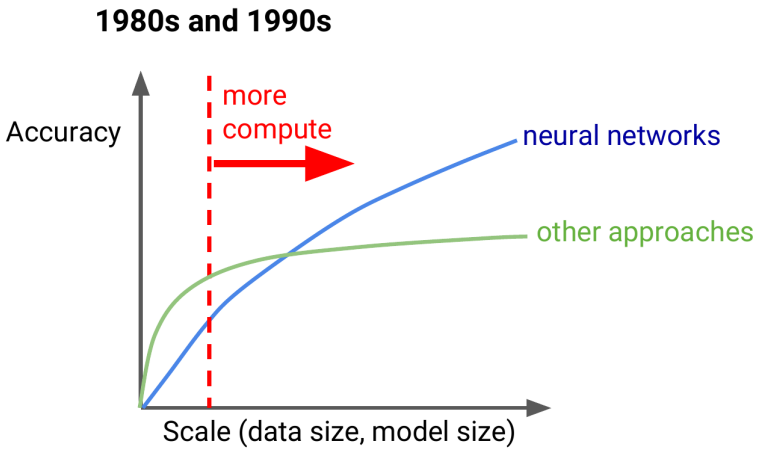
- Massive amount of training dataset
- Large number of parameters

Accuracy vs. Data/Model Size



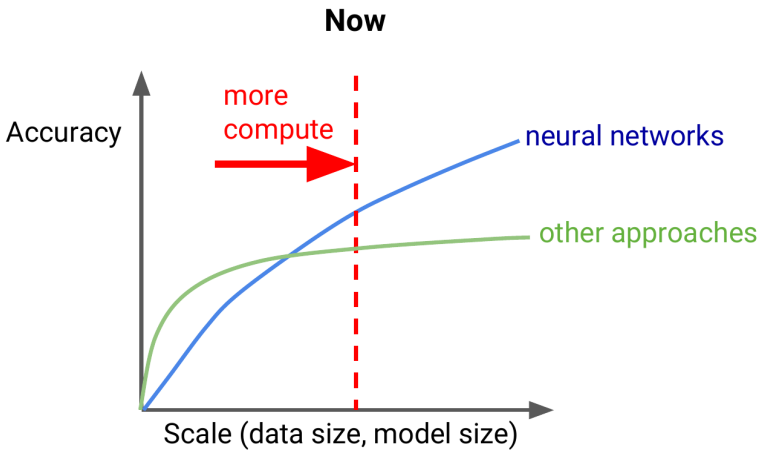
[Jeff Dean at AI Frontiers: Trends and Developments in Deep Learning Research]

Accuracy vs. Data/Model Size



[Jeff Dean at AI Frontiers: Trends and Developments in Deep Learning Research]

Accuracy vs. Data/Model Size



[Jeff Dean at AI Frontiers: Trends and Developments in Deep Learning Research]

Why Does Deep Learning Work Now?

- Huge quantity of data
- Tremendous increase in computing power
- Better training algorithms

Data

GPUs

Weight Initialization

Non-Linearity

Linear Algebra Review

Vector

- A vector is an array of numbers.
- Notation:
  - Denoted by **bold** lowercase letters, e.g., $\mathbf{x}$.
  - x_i denotes the ith entry.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Matrix and Tensor

- A matrix is a 2-D array of numbers.
- A tensor is an array with more than two axes.
- Notation:
  - Denoted by **bold** uppercase letters, e.g., $\mathbf{A}$.
  - $\mathtt{a}_{\mathtt{ij}}$ denotes the entry in $\mathtt{i}$th row and $\mathtt{j}$th column.
  - If $\mathbf{A}$ is $\mathtt{m} \times \mathtt{n}$, it has $\mathtt{m}$ rows and $\mathtt{n}$ columns.

$$\mathbf{A} = \begin{bmatrix} \mathtt{a}_{1,1} & \mathtt{a}_{1,2} & \mathtt{a}_{1,3} & \dots & \mathtt{a}_{1,\mathtt{n}} \\ \mathtt{a}_{2,1} & \mathtt{a}_{2,2} & \mathtt{a}_{2,3} & \dots & \mathtt{a}_{2,\mathtt{n}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathtt{a}_{\mathtt{m},1} & \mathtt{a}_{\mathtt{m},2} & \mathtt{a}_{\mathtt{m},3} & \dots & \mathtt{a}_{\mathtt{m},\mathtt{n}} \end{bmatrix}$$

Matrix Addition and Subtraction

- The matrices must have the same dimensions.

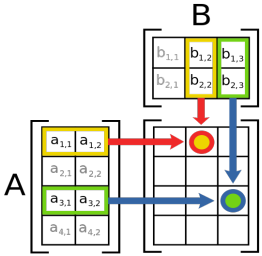
$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

Matrix Product

- The matrix product of matrices **A** and **B** is a third matrix **C**, where $\mathbf{C} = \mathbf{AB}$.
- If **A** is of shape $\mathtt{m} \times \mathtt{n}$ and **B** is of shape $\mathtt{n} \times \mathtt{p}$, then **C** is of shape $\mathtt{m} \times \mathtt{p}$.

$$\mathtt{c_{ij}} = \sum_{\mathtt{k}} \mathtt{a_{ik}} \mathtt{b_{kj}}$$

- Properties
  - Associative: $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$
  - Not commutative: $\mathbf{AB} \neq \mathbf{BA}$



[https://en.wikipedia.org/wiki/Matrix_multiplication]

Matrix Transpose

- Swap the rows and columns of a matrix.

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \Rightarrow \mathbf{A}^\intercal = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

- Properties
  - $\mathbf{A}_{ij} = \mathbf{A}^\intercal_{ji}$
  - If $\mathbf{A}$ is $m \times n$, then $\mathbf{A}^\intercal$ is $n \times m$
  - $(\mathbf{A} + \mathbf{B})^\intercal = \mathbf{A}^\intercal + \mathbf{B}^\intercal$
  - $(\mathbf{AB})^\intercal = \mathbf{B}^\intercal\mathbf{A}^\intercal$

Inverse of a Matrix

- If $\mathbf{A}$ is a square matrix, its inverse is called $\mathbf{A}^{-1}$.

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

- Where $\mathbf{I}$, the identity matrix, is a diagonal matrix with all 1's on the diagonal.

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

L^p Norm for Vectors

- We can measure the size of vectors using a norm function.
- Norms are functions mapping vectors to non-negative values.
- L^1 norm

$$||\mathbf{x}||_1 = \sum_i |x_i|$$

- L^2 norm

$$||\mathbf{x}||_2 = (\sum_i |x_i|^2)^{\frac{1}{2}} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

- L^p norm

$$||\mathbf{x}||_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$$

Probability Review

Random Variables

- Random variable: a variable that can take on different values randomly.

- Random variables may be discrete or continuous.
  - Discrete random variable: finite or countably infinite number of states
  - Continuous random variable: real value

- Notation:
  - Denoted by an upper case letter, e.g., $\mathtt{X}$
  - Values of a random variable $\mathtt{X}$ are denoted by lower case letters, e.g., $\mathtt{x}$ and $\mathtt{y}$.

Probability Distributions

- Probability distribution: how likely a random variable is to take on each of its possible states.
  - E.g., the random variable `X` denotes the outcome of a coin toss.
  - The probability distribution of `X` would take the value `0.5` for `X = head`, and `0.5` for `Y = tail` (assuming the coin is fair).

- The way we describe probability distributions depends on whether the variables are discrete or continuous.

Discrete Variables

- Probability mass function (PMF): the probability distribution of a discrete random variable $\mathtt{X}$.
- Notation: denoted by a lowercase $\mathtt{p}$.
  - E.g., $\mathtt{p}(\mathtt{x}) = 1$ indicates that $\mathtt{X} = \mathtt{x}$ is certain
  - E.g., $\mathtt{p}(\mathtt{x}) = 0$ indicates that $\mathtt{X} = \mathtt{x}$ is impossible
- Properties:
  - The domain $\mathtt{D}$ of $\mathtt{p}$ must be the set of all possible states of $\mathtt{X}$
  - $\forall \mathtt{x} \in \mathtt{D}(\mathtt{X}), 0 \leq \mathtt{p}(\mathtt{x}) \leq 1$
  - $\sum_{\mathtt{x} \in \mathtt{D}(\mathtt{X})} \mathtt{p}(\mathtt{x}) = 1$

Independence

- Two random variables $\mathtt{X}$ and $\mathtt{Y}$ are independent, if their probability distribution can be expressed as their products.

$$\forall \mathtt{x} \in \mathtt{D(X)}, \mathtt{y} \in \mathtt{D(Y)}, \mathtt{p(X = x, Y = y)} = \mathtt{p(X = x)p(Y = y)}$$

- E.g., if a coin is tossed and a single 6-sided die is rolled, then the probability of landing on the head side of the coin and rolling a 3 on the die is:

$$\mathtt{p(X = head, Y = 3)} = \mathtt{p(X = head)p(Y = 3)} = \frac{1}{2} \times \frac{1}{6} = \frac{1}{12}$$

Conditional Probability

- Conditional probability: the probability of an event given that another event has occurred.

$$\mathtt{p}(\mathtt{Y} = \mathtt{y} \mid \mathtt{X} = \mathtt{x}) = \frac{\mathtt{p}(\mathtt{Y} = \mathtt{y}, \mathtt{X} = \mathtt{x})}{\mathtt{p}(\mathtt{X} = \mathtt{x})}$$

- E.g., if 60% of the class passed both labs and 80% of the class passed the first labs, then what percent of those who passed the first lab also passed the second lab?
  - E.g., $\mathtt{X}$ and $\mathtt{Y}$ random variables for the first and the second labs, respectively.

$$\mathtt{p}(\mathtt{Y} = \mathtt{lab2} \mid \mathtt{X} = \mathtt{lab1}) = \frac{\mathtt{p}(\mathtt{Y} = \mathtt{lab2}, \mathtt{X} = \mathtt{lab1})}{\mathtt{p}(\mathtt{X} = \mathtt{lab1})} = \frac{0.6}{0.8} = \frac{3}{4}$$

Expectation

- The expected value of a random variable X with respect to a probability distribution $p(X)$ is the average value that X takes on when it is drawn from $p(X)$.

$$E_{x \sim p}[X] = \sum_{x} p(x)x$$

- E.g., If $X : \{1, 2, 3\}$, and $p(X = 1) = 0.3$, $p(X = 2) = 0.5$, $p(X = 3) = 0.2$
  - $E[X] = 0.3 \times 1 + 0.5 \times 2 + 0.2 \times 3 = 1.9$

Variance and Standard Deviation

- The variance gives a measure of how much the values of a random variable $\mathtt{X}$ vary as we sample it from its probability distribution $\mathtt{p(X)}$.

$$\mathtt{Var(X)} = \mathtt{E}[(\mathtt{X} - \mathtt{E[X]})^2]$$
$$\mathtt{Var(X)} = \sum_{\mathtt{x}} \mathtt{p(x)}(\mathtt{x} - \mathtt{E[X]})^2$$

- E.g., If $\mathtt{X} : \{1, 2, 3\}$, and $\mathtt{p(X = 1)} = 0.3$, $\mathtt{p(X = 2)} = 0.5$, $\mathtt{p(X = 3)} = 0.2$
  - $\mathtt{E[X]} = 0.3 \times 1 + 0.5 \times 2 + 0.2 \times 3 = 1.9$
  - $\mathtt{Var(X)} = 0.3(1 - 1.9)^2 + 0.5(2 - 1.9)^2 + 0.2(3 - 1.9)^2 = 0.49$

- The standard deviation, shown by σ, is the square root of the variance.

Covariance (1/2)

- The covariance gives some sense of how much two values are linearly related to each other.

$$\mathtt{Cov}(\mathtt{X},\mathtt{Y}) = \mathtt{E}[(\mathtt{X}-\mathtt{E}[\mathtt{X}])(\mathtt{Y}-\mathtt{E}[\mathtt{Y}])]$$

$$\mathtt{Cov}(\mathtt{X},\mathtt{Y}) = \sum_{(\mathtt{x},\mathtt{y})}\sum \mathtt{p}(\mathtt{x},\mathtt{y})(\mathtt{x}-\mathtt{E}[\mathtt{X}])(\mathtt{y}-\mathtt{E}[\mathtt{Y}])$$

Covariance (2/2)

			Y		
	p(X, Y)	1	2	3	p(X)
	1	1/4	1/4	0	1/2
X	2	0	1/4	1/4	1/2
	p(Y)	1/4	1/2	1/4	1

$$\mathrm{E}[X] = \frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2} \qquad \mathrm{E}[Y] = \frac{1}{4} \times 1 + \frac{1}{2} \times 2 + \frac{1}{4} \times 3 = 2$$

$$\mathrm{Cov}(X, Y) = \sum_{(x,y)} \sum p(x, y)(x - \mathrm{E}[X])(y - \mathrm{E}[Y])$$

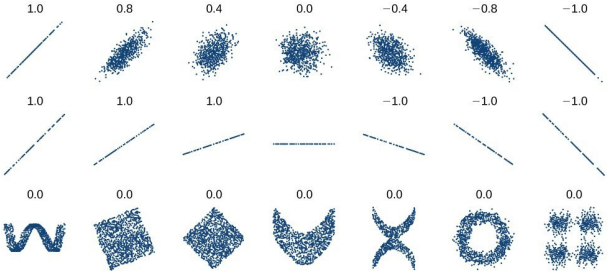
$$= \frac{1}{4}(1 - \frac{3}{2})(1 - 2) + \frac{1}{4}(1 - \frac{3}{2})(2 - 2) + 0(1 - \frac{3}{2})(3 - 2)$$

$$+0(2 - \frac{3}{2})(1 - 2) + \frac{1}{4}(2 - \frac{3}{2})(2 - 2) + \frac{1}{4}(2 - \frac{3}{2})(3 - 2) = \frac{1}{4}$$

Correlation Coefficient

- The Correlation coefficient is a quantity that measures the strength of the association (or dependence) between two random variables, e.g., $\mathtt{X}$ and $\mathtt{Y}$.

$$\rho(\mathtt{X},\mathtt{Y}) = \frac{\mathtt{Cov}(\mathtt{X},\mathtt{Y})}{\sigma(\mathtt{X})\sigma(\mathtt{Y})}$$



Probability and Likelihood (1/2)

- Let $\mathtt{X} : \{\mathtt{x}^{(1)}, \mathtt{x}^{(2)}, \cdots, \mathtt{x}^{(\mathtt{m})}\}$ be a discrete random variable drawn independently from a distribution probability $\mathtt{p}$ depending on a parameter θ.
  - For six tosses of a coin, $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$, $\mathtt{h}$: head, and $\mathtt{t}$: tail.
  - Suppose you have a coin with probability θ to land heads and $(1 - \theta)$ to land tails.

- $\mathtt{p}(\mathtt{X} \mid \theta = \frac{2}{3})$ is the probability of $\mathtt{X}$ given $\theta = \frac{2}{3}$.

- $\mathtt{p}(\mathtt{X} = \mathtt{h} \mid \theta)$ is the likelihood of θ given $\mathtt{X} = \mathtt{h}$.

- Likelihood ($\mathtt{L}$): a function of the parameters (θ) of a probability model, given specific observed data, e.g., $\mathtt{X} = \mathtt{h}$.

$$\mathtt{L}(\theta \mid \mathtt{X}) = \mathtt{p}(\mathtt{X} \mid \theta)$$

Probability and Likelihood (2/2)

- The likelihood differs from that of a probability.
- A probability $\mathtt{p}(\mathtt{X} \mid \theta)$ refers to the occurrence of future events.
- A likelihood $\mathtt{L}(\theta \mid \mathtt{X})$ refers to past events with known outcomes.

Maximum Likelihood Estimator

- If samples in $\mathtt{X}$ are independent we have:

$$\begin{aligned}\mathtt{L}(\theta \mid \mathtt{X}) = \mathtt{p}(\mathtt{X} \mid \theta) &= \mathtt{p}(\mathtt{x}^{(1)}, \mathtt{x}^{(2)}, \cdots, \mathtt{x}^{(\mathtt{m})} \mid \theta) \\ &= \mathtt{p}(\mathtt{x}^{(1)} \mid \theta)\mathtt{p}(\mathtt{x}^{(2)} \mid \theta) \cdots \mathtt{p}(\mathtt{x}^{(\mathtt{m})} \mid \theta) = \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)\end{aligned}$$

- The maximum likelihood estimator (MLE): what is the most likely value of θ given the training set?

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \arg\max_{\theta} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

Maximum Likelihood Estimator - Example

- Six tosses of a coin, with the following model:
  - Possible outcomes: h with probability of θ, and t with probability $(1-\theta)$.
  - Results of coin tosses are independent of one another.
- Data: $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$
- The likelihood is

$$\begin{aligned}\mathtt{L}(\theta \mid \mathtt{X}) &= \mathtt{p}(\mathtt{X} \mid \theta) \\ &= \mathtt{p}(\mathtt{X}=\mathtt{h} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{h} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta) \\ &= \theta(1-\theta)(1-\theta)(1-\theta)\theta(1-\theta) \\ &= \theta^2(1-\theta)^4\end{aligned}$$

- $\hat{\theta}$ is the value of θ that maximizes the likelihood:

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \frac{2}{2+4}$$

Log-Likelihood

- The MLE product is prone to numerical underflow.

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \arg\max_{\theta} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

- To overcome this problem we can use the logarithm of the likelihood.
  - It does not change its arg max, but transforms a product into a sum.

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{log}\mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

Negative Log-Likelihood

- Likelihood: $\mathtt{L}(\theta \mid \mathtt{X}) = \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Log-Likelihood: $\mathtt{logL}(\theta \mid \mathtt{X}) = \mathtt{log} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta) = \sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{logp}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Negative Log-Likelihood: $-\mathtt{logL}(\theta \mid \mathtt{X}) = -\sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{logp}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Negative log-likelihood is also called the cross-entropy

Cross-Entropy

- Coss-entropy: quantify the difference (error) between two probability distributions.
- How close is the predicted distribution to the true distribution?

$$H(p, q) = -\sum_{x} p(x) \log(q(x))$$

- Where p is the true distribution, and q the predicted distribution.

Cross-Entropy - Example

- Six tosses of a coin: $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$
- The true distribution $\mathtt{p}$: $\mathtt{p}(\mathtt{h}) = \frac{2}{6}$ and $\mathtt{p}(\mathtt{t}) = \frac{4}{6}$
- The predicted distribution $\mathtt{q}$: $\mathtt{h}$ with probability of θ, and $\mathtt{t}$ with probability $(1 - \theta)$.
- Cross entropy: $\mathtt{H}(\mathtt{p}, \mathtt{q}) = -\sum_{\mathtt{x}} \mathtt{p}(\mathtt{x})\mathtt{log}(\mathtt{q}(\mathtt{x}))$
  $= -\mathtt{p}(\mathtt{h})\mathtt{log}(\mathtt{q}(\mathtt{h})) - \mathtt{p}(\mathtt{t})\mathtt{log}(\mathtt{q}(\mathtt{t})) = -\frac{2}{6}\mathtt{log}(\theta) - \frac{4}{6}\mathtt{log}(1 - \theta)$
- Likelihood: $\theta^2(1 - \theta)^4$
- Negative log likelihood: $-\mathtt{log}(\theta^2(1 - \theta)^4) = -2\mathtt{log}(\theta) - 4\mathtt{log}(1 - \theta)$

References

- Ian Goodfellow et al., Deep Learning (Ch. 1, 2, 3)

Questions?

Acknowledgements

Some of the pictures were copied from the book Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurelien Geron, O'Reilly Media, 2017.